REMOTE SERVER OBJECT ARCHITECTURE FOR SPEECH RECOGNITION

Field of the Invention

The present invention relates generally to a remote server object architecture for speech recognition and more particularly to a speech recognition system including a number of remote server objects which enable the speech recognition system to transparently process incoming voice data over a number of computer systems within a network.

Cross References To Related Applications

This application claims the benefit of priority from commonly owned U.S. Provisional Patent Application Serial Number 60/192,091, filed March 24, 2000, entitled COMBINED SYNTACTIC AND SEMANTIC SEARCH, PARSING, AND APPLICATION ACCESS; U.S. Provisional Patent Application Serial Number 60/191,915, filed March 24, 2000, entitled SPEECH RECOGNITION APPLICATION TECHNOLOGY USING WEB, SCRIPTING AND SEMANTIC OBJECTS; U.S. Provisional Patent Application Serial Number 60/192,090, filed March 24, 2000, entitled A NOVEL APPROACH TO SPEECH RECOGNITION; and U.S. Provisional Patent Application Serial Number 60/192,076, filed March 24, 2000, entitled REMOTE SERVER OBJECT ARCHITECTURE FOR SPEECH RECOGNITION.

This application is also related to the following copending U.S. patent applications, the contents of which are incorporated herein in their entirety by reference:

"A Novel Approach To Speech Recognition," U.S. Patent Application Serial Number _____, attorney docket number ELZ-1;

"Phonetic Data Processing System and Method," U.S. Patent Application Serial Number _____, attorney docket number ELZ-2; and

"Web-Based Speech Recognition With Scripting and Semantic Objects," U.S. Patent Application Serial Number _____, attorney docket number ELZ-4.

Background of the Invention

In the new, connected economy, it has become increasingly important for companies or service providers to become more in tune with their clients and customers. Such contact can be facilitated with automated telephonic transaction systems, in which interactively-generated prompts are played in the context of a telephone transaction, and the replies of a human user are recognized by a speech recognition system. The answers given by the respondent are processed by the system in order to convert the spoken words to meaning, which can then be utilized interactively, or stored in a database.

In order for a computer system to recognize the words that are spoken and convert these words to text, the system must be programmed to phonetically break down the words and convert portions of the words to their textural equivalents. Such a conversion requires an understanding of the components of speech and the formation of the spoken word. The production of speech generates a complex series of rapidly changing pressure waveforms. These waveforms comprise the basic building blocks of speech, known as phonemes. Vowel and consonant

sounds are made up of phonemes and have many different characteristics, depending on which components of human speech are used. The position of a phoneme in a word has a significant effect on the ultimate sound generated. A spoken word can have several meanings, depending on how it is said. Speech scientists have identified allophones as acoustic variants of phonemes and use them to more explicitly define how a particular word is formed.

While there are several distinct methods for analyzing the spoken word and extracting the information necessary to enable the recognition system to convert the speech to word-strings, including Hidden Markov modeling and neural networks, these methods generally perform similar operations. The differences in these methods are typically in the manner in which the system determines how to break the phonetic signal into portions that define phonemes. Generally, a speech recognition system first converts an incoming analog voice signal into a digital signal. The second step is called feature extraction, wherein the system analyzes the digital signal to identify the acoustic properties of the digitized signal. Feature extraction generally breaks the voice down into its individual sound components. Conventional techniques for performing feature extraction include subband coding Fast Fourier Transforms and Linear Predictive Coding. Once the signal has been analyzed, the system then determines where distinct acoustic regions occur. The goal of this step is to divide the acoustic signal into regions that will be identified as phonemes which can be converted to a textural format. In isolated word systems, this process is simplified, because there is a pause after each word. In continuous speech systems, however, this process is much more difficult, since there typically are no breaks between words in the acoustic stream. Accordingly, the system must be able not only to break the words themselves into distinct acoustic regions, but must also be able to separate consecutive words in the stream. It is in this step that conventional methods such as Hidden Markov modeling and neural networks are used. The

final step involves comparing a specific acoustic region, as determined in the previous step, to a known set of templates in a database in order to determine the word or word portion represented by the acoustic signal region. If a match is found, the resulting textural word is output from the system. If one is not, the signal can either be dynamically manipulated in order to increase the chances of finding a match, or the data can be discarded and the system prompted to repeat the question to the respondent, if the associated answer cannot be determined due to the loss of the data.

Summary of the Invention

The present invention is directed to implementation of computing-intensive speech recognition systems that require simultaneous operations of multiple functional units. This is accomplished by having a number of discrete servers which can be located in separate computer systems. Four types of functions related to speech recognition have been identified and isolated in different operating units. Performance and maintenance is enhanced by having a unidirectional data pipeline, from one server to another, while control flow streams in the opposite direction.

The configuration of the system provides the ability to separate and encapsulate the modular high-level functions associated with each server, which are controlled by a central monitoring system. Consequently, the system is capable of enabling changes to transactions and maintenance on the system without shutting down the entire system and is able to compensate for malfunctions in any one or more of the components of the system without affecting the operation of the entire system. System-wide load balancing is made possible by means of borrowing functional servers from the applications having low load and reassigning them to more demanding applications.

According to one aspect of the invention, a speech recognition system includes a line of service including a first server object coupled to a telephone network for receiving a voice data message from the telephone network, a second server object having a first connection to the first server object for receiving the voice data message from the first server object and converting the voice data message to a phonetic data message, a third server object having a second connection to the second server object for receiving the phonetic data message from the second server object and converting the phonetic data message to a syntactic data message and a fourth server object having a third connection to the third server object for receiving the syntactic data message from the third server object and converting the syntactic data message to a semantic data message, which is representative of the voice data message. The first, second, third and fourth server objects may be remote with respect to each other and the first, second and third connections are formed over a first computer network.

The fourth server object may be coupled to a second computer network for receiving an application code from a client of the second computer network, the application code providing control data for the operation of the speech recognition system. The first computer and the second computer network may be one of a local area network and the internet. The first, second and third connections may be formed from named pipes. The system may further include a control monitor for controlling the configuration of the first, second, third and fourth server objects in the line of service. At least one of the first, second, third and fourth server objects periodically may transmit a status signal to the system monitor, wherein the transmission of the periodic status signal from the at least one of the first, second, third and fourth server objects to the system monitor indicates that the one of the first, second, third and fourth server objects is operational. A nontransmission of the periodic status signal from the at least one of the first, second, third and fourth server objects to the system monitor indicates that the one

of the first, second, third and fourth server objects is disabled. The system may further include at least one backup server object which is configured into the system by the system monitor when the at least one of the first, second, third and fourth server objects is disabled. The first, second, third and fourth server objects are configured by the system monitor according to the Distributed Component Object Model (DCOM). Each of the first, second, third and fourth server objects may include a post office for addressing and routing the voice data message, the phonetic data message, the syntactic data message and the semantic data message through the line of service from the telephone network to the second computer network. The system may further include additional lines of service connected between the telephone network and the second computer network.

According to another aspect of the invention, a method of processing speech includes:

A. receiving, at a first server object, a voice data message from a telephone network;

B. transmitting the voice data message over a first computer network to a second server object;

C. converting the voice data message to a phonetic data message in the second server object;

D. transmitting the phonetic data message from the second server object to a third server object over the first computer network;

E. converting the phonetic data message to a syntactic data message in the third server object;

F. transmitting the syntactic data message from the third server object to a fourth server object over the first computer network; and

G. converting the syntactic data message to a semantic data message representative of the voice data message in the fourth server object.

The method may further include downloading an application code from a second computer network, wherein the application code includes control code for controlling operation of the first, second, third and fourth server objects. The first computer network and the second computer network is one of a local area network and the internet. Steps B, D and F may include transmitting the voice data message, the phonetic data message and the syntactic data message, respectively, over named pipes. The method may further include configuring the first, second, third and fourth server objects with a control monitor. The method may further include the step of at least one of the first, second, third and fourth server objects transmitting a status signal to the control monitor, wherein the transmission of the periodic status signal from the at least one of the first, second, third and fourth server objects to the control monitor indicates that the at least of the first, second, third and fourth server objects is operational. The nontransmission of the periodic status signal from the at least one of the first, second, third and fourth server objects to the control monitor may indicate that the at least of the first, second, third and fourth server objects is disabled. the first, second, third and fourth server objects are configured by the system monitor according to the Distributed Component Object Model (DCOM). The method may further include processing the semantic data message in the fourth server object according to the application code.

According to yet another aspect of the invention, a speech recognition system includes a line of service having a voice server object coupled to a telephone network for receiving a voice data message from the telephone network, a speech recognition server having a connection to the voice server object for receiving the voice data message from the voice server object and converting the voice data message to a phonetic data message and converting the phonetic data message to a syntactic data message and a task server object having a connection to the speech recognition server for receiving the syntactic data

message from the speech recognition server object and converting the syntactic data message to a semantic data message, which is representative of the voice data message. The connections between the voice server object, the speech recognition server and the task server object are formed over a first computer network.

The task server object may be coupled to a second computer network for receiving an application code from a client of the second computer network, the application code providing control data for the operation of the speech recognition system. The first computer network and the second computer network may be one of a local area network and the internet, and the connections may be formed from named pipes. The system may further include a control monitor for controlling the configuration of the voice server object, the speech recognition server and the task server object in the line of service. At least one of the voice server object, the speech recognition server and the task server object may periodically transmit a status signal to the system monitor, wherein the transmission of the periodic status signal from the at least one of the voice server object, the speech recognition server and the task server object to the system monitor indicates that the one of the voice server object, the speech recognition server and the task server object is operational. A nontransmission of the periodic status signal from the at least one of the voice server object, the speech recognition server and the task server object to the system monitor may indicate that the at least one of the voice server object, the speech recognition server and the task server object is disabled. The system may further include at least one backup server object which is configured into the system by the system monitor when the at least one of the voice server object, the speech recognition server and the task server object is disabled. The voice server object, the speech recognition server and the task server object may be configured by the system monitor according to the Distributed Component Object Model (DCOM). Each of the

voice server object, the speech recognition server and the task server object may include a post office for addressing and routing the voice data message, the phonetic data message, the syntactic data message and the semantic data message through the line of service from the telephone network to the second computer network. The system may further include additional lines of service connected between the telephone network and the second computer network. The speech recognition server may include an acoustic server object for receiving the voice data message from the voice server object and converting the voice data message to the phonetic data message and a symbolic server object for receiving the phonetic data message from the acoustic server object and converting the phonetic data message to the syntactic data message. The voice, acoustic, symbolic and task server objects are remote with respect to each other.

Brief Description Of The Drawings

The foregoing and other objects of this invention, the various features thereof, as well as the invention itself may be more fully understood from the following description when read together with the accompanying drawings in which:

Fig. 1 is a schematic block diagram of the remote server object architecture for speech recognition system in accordance with the present invention;

Fig. 2 is a schematic block diagram of the remote server objects associated with one line of service in accordance with the present invention;

Fig. 3 is a schematic block diagram of an array of remote server objects in accordance with the present invention; and

Fig. 4 is a schematic block diagram showing the hardware configuration of the system in accordance with the present invention.

Detailed Description

The present invention is directed to a speech recognition system which can be used to conduct automated telephone transactions. In general, the system receives an application over either a local network or the internet from a party conducting the transaction. The application contains the code that operates the transaction. The transaction prompts are presented to the respondent over a telephone network, and the replies are received by the speech recognition system spoken by the respondent. The voice data that represent the answers given by the respondent are broken down into a phonetic stream of data which can be recognized by the system. The system converts the phonetic stream of data into a word list or syntactic data message which is then converted to a semantic representation of the spoken answers given by the respondent. The semantic representation can then be transmitted to a transaction initiator over the local network or the internet.

Fig. 1 is a schematic block diagram of the speech recognition system 10 of the present invention. The system 10 includes a voice server 12, an acoustic server 14, a symbolic server 16 and a task server 18, all of which are connected to a network, such as a LAN, and which operate under the control of system monitor 20. Servers 12, 14, 16 and 18 are configured as separate Remote Server Objects (RSO) which are located on separate computer systems and which perform a different function on the data stream. As shown in Fig. 1, servers 12, 14, 16 and 18 form a single line of service 26 between a telephone network 22 and the internet 24. As is discussed below, a plurality of lines of service can be implemented similar to the line 26.

In general, the voice server 12 receives the audio voice stream from the telephone network 22, acoustic server 14 converts the audio voice stream into an output phonetic stream, symbolic server 16 converts the phonetic stream to a syntactic data message and task server converts the syntactic data message output from the symbolic sever 16 into a semantic representation of the original input audio voice stream. The functions of the servers 12, 14, 16 and 18 are discussed in greater detail below.

Each RSO is implemented as a discrete object according to the Distributed Component Object Model (DCOM). DCOM is a set of Microsoft® concepts and program interfaces in which a client program object can request services from server program objects on other computers in a network. Configuring each RSO as a DCOM object enables each RSO to perform its operation on the data being input to the system before the results of the operation are passed to the next RSO for further processing. Accordingly, each RSO in the system is capable of performing its specialized high-level function in a modular fashion before passing the necessary data to the next RSO.

By forming the RSO architecture as DCOM objects, each object in the system 10 can be configured from different implementations, technologies, hardware and system configurations, all within the same system architecture. The high-level functions carried out by each RSO remain independent of the implementation details of the underlying architecture and therefore enable the system to be operated in a transparent manner from the point of view of the party conducting the transaction and the respondent. Furthermore, the implementation of the system independent of the underlying architecture allows re-use of common system components, which allows a maximum of tested, reliable software to be used. Furthermore, separable functions allow the modular components to be tested in isolation, thus simplifying maintenance and support.

As shown in Fig. 1, each RSO 12, 14, 16 and 18 include a plurality of data lines D and control lines C connected therebetween. As is described below, the connection between RSO is via named pipes. The lines D and C in Fig. 1 are simply schematic representations of the flow of data in the system 10. The system is a context independent, data feed forward design, which means that voice data flows in one direction only, from the telephone network 22 to the internet 24. Control data flows from the application downloaded by the task server 18 from the internet 24 in the opposite direction of the voice data in order to enable the application to control the RSO's according to the specific code in the application.

The flow of voice data and control data will now be discussed with reference to Fig. 2. Shown in Fig. 2 is a schematic block diagram of the line of service 26 of Fig. 1. Line 26 includes a first unit 40, which preferably is a computer running the Windows NT server operating system, including voice server 12, which is coupled to post office unit 44 by CRcvr interface 46. Voice server 12 is connected to telephone network 22 preferably by a third party DS-1 interface 42. Second unit 50, which preferably is a computer running the Windows NT workstation operating system, includes acoustic server 14, which is coupled to post office unit 54 by CRcvr interface 52, and symbolic server 16, which is coupled to post office unit 56 by CRcvr interface 58. Third unit 60, which preferably is a computer running the Windows NT server operating system, includes task server 18, which is coupled to post office unit 64 by CRcvr interface 66. The CRcvr interfaces 46, 52, 58 and 66 are message-based communication interfaces which enable the server objects 12, 14, 16 and 18 to pass voice data and control data therebetween. The simplified code for the CRcvr interface is:

```
Interface CRcvr
{
void Idle ();
BOOL Receive (CMsg& msg);
};
```

where the CMsg object provides a wrapper around an array of data, which, in the case of the present invention, is the voice data message. The CMsg object adds to the voice data message a message header, sender ID, receiver ID, message ID and optional arguments.

A set of named pipes 80 is used to allow the flow of data between the post offices associated with each of the servers 12, 14, 16 and 18. A named pipe is a method for passing information from one computer process to other processes using a pipe or message holding place that is given a specific name. Unlike a regular pipe, a named pipe can be used by processes that do not have to share a common process origin and the message sent to the named pipe can be read by any authorized process that knows the name of the named pipe. As shown in Fig. 2, voice data flows only in the direction of arrow 70 and control data flows only in the direction of arrow 72, thus ensuring correct synchronization at all times.

In operation, the voice server 12 receives a voice data message in the form of a voice stream from the telephone network 22 over voice interface 42. The CRcvr interface 46 applies the appropriate headings on the voice data message and forwards the message to post office 44, which routes the message, based on the receiver ID added by the CMsg object, to the post office 54, via the named pipe 80. In addition to receiving the voice data message from the telephone network, voice server 12 controls all incoming and outgoing telephone calls, audio playback and recording of the transaction questions and voice prompt handling. It also performs system load and stress testing.

Acoustic server 14 receives the voice data message and converts the voice data message to a phonetic data message. The method implemented by acoustic server 14 for converting voice data to phonetic data is described in commonly assigned copending U.S. Patent Application No. _____ (Attorney Docket No. ELZ-1) entitled A NOVEL APPROACH TO SPEECH RECOGNITION, which application is herein incorporated by reference in its entirety.

After the acoustic server 14 has converted the voice data message to a phonetic data message , CRvcr 52 attaches the appropriate header to the phonetic data message and transfers it to post office 56 via post office 54 and named pipe 80. Symbolic server 16 receives the phonetic data message from post office 56 via CRcvr interface 58 and converts the phonetic data message to a syntactic data message, based on the method described in commonly assigned copending copending U.S. Patent Application No. _____ (Attorney Docket No. ELZ-2) entitled COMBINED SYNTACTIC AND SEMANTIC SEARCH, PARSING, AND APPLICATION, which application is herein incorporated by reference in its entirety.

Prior to the commencement of the transaction, Task server 18 initiates a connection to the internet 24 over connection 62 and downloads the transaction application code from the transaction initiator's website. The task server operates under the control of the application code to conduct the transaction defined in the code. Based on the application code, task server 18 controls the operation of the symbolic server 16 which, in turn, controls the operation of the acoustic server 14 and the voice server 12. All control data is transmitted in the direction of arrow 72 only.

Task server 18 receives the syntactic data message from the symbolic server via post office 56, named pipe 80, post office 64 and CRcvr interface 66. Task server 18 then converts the syntactic data message to a semantic

representation of the original syntactic data message and processes the semantic data according to the application code.

In addition to the voice server 12, acoustic server 14, symbolic server 16 and task server 18, the line 26 may include a synthesizing server (not shown) for implementing text-to-speech synthesizers, for converting incoming text messages to audio streaming messages. The synthesizing server could be implemented when conducting transactions to hearing impaired respondents, for example. Furthermore, a first recording server could be associated with the symbolic server 16 for recording data flow from the symbolic server 16. Such data, which includes all the information required to reconstruct all application events, may be used to debug the application code. A second recording server may be associated with the voice server for recording the input voice data for later use in system development and quality-control purposes.

While the system has been described as a single line of service 26, the system may be configured as a multiple-line system 110, as shown in Fig. 3. System 110 comprises a system having 24 lines of service, which enable the system 110 to conduct 24 different transactions simultaneously. Each line of service includes a voice server 112, an acoustic server 114, a symbolic server 116 and a task server 118. The configuration and operation of each line of service in system 110 is identical to the configuration and operation of line of service 26 described above. As shown in Fig. 4, all 24 voice servers 112 are grouped in a single Windows NT server PC 130 and all 24 task servers 118 are grouped in a single Windows NT server PC 132. A pair of acoustic servers and their associated symbolic servers are grouped within each of twelve Windows NT workstation PC's 134.

While the example described herein and shown in the figures depicts 24 lines of service, it will be understood that any reasonable number of lines of service may be implemented with the present invention. Furthermore, one or

more of the servers may be used as backup servers in the event that one or more of the active servers becomes disabled. Since the system is network-based, any one of the severs can be replaced with a backup server simply by rerouting the necessary voice data and control data through the new server instead of the disabled server.

As set forth above, Each of the RSO's 12, 14, 16 and 18 are under the control of system monitor 20, Fig. 1. System monitor 20 operates under the control of a configuration file loaded into the system monitor by the administrator of the system 10. The system monitor 20 reads the configuration file and creates lines of service 26 according to the description contained in the file. It also creates and maintains a list of backup servers of each of the different types. Each of the RSO's in each line of service 26 created by the system monitor 20 is identified by DCOM programmatic identifiers. The system monitor 20 also assigns and registers post offices to each RSO and establishes named message pipes 80 for direct RSO-to-RSO high speed communication, as shown in Fig. 2. Once each post office is configured, each RSO has a direct connection to all other RSO's in the line.

In order to ensure that each RSO is operating properly, each RSO outputs a "heartbeat" message that is detected by the system monitor 20. As long as the heartbeat continues, the system monitor determines that each RSO in the system 10 is operating properly. If the system monitor 20 does not detect a heartbeat for a predetermined period of time, the system monitor determines that a particular RSO is disabled and configures a backup RSO to take the place of the disabled RSO. The replacement is handled such that the effected line of service is put out of operation until the repair is complete, so that the failure of the RSO affects only the current transaction. As soon as the system monitor 20 reconfigures the operating RSO's with a backup RSO for the disabled RSO, the line of service is operational. The system monitor is also capable of maintaining system-wide load

balancing by reassigning RSO's in low load applications to applications having a greater demand for resources.

Accordingly, the present invention provides a system for and method of conducting telephonic transactions including a speech recognition system in which the architecture of the system is transparent to the respondent and to the party conducting the transaction. The system includes a plurality of remote server objects that are connected to each other over a network. The voice server RSO is connected to a telephone network for communication with the respondent. The task server RSO is connected to the internet and downloads the application code for a particular transaction from the website of the transaction initiator. The voice data message received by the voice server is processed by the system and is converted to a semantic representation of the voice data. Once the data is converted, it is applied to the application code to be processed. Since the architecture is transparent to the application, all maintenance, testing and support can be carried out without affecting or disrupting the active transactions.

The invention may be embodied in other specific forms without departing from the spirit or essential characteristics thereof. The present embodiments are therefore to be considered in respects as illustrative and not restrictive, the scope of the invention being indicated by the appended claims rather than by the foregoing description, and all changes which come within the meaning and range of the equivalency of the claims are therefore intended to be embraced therein.